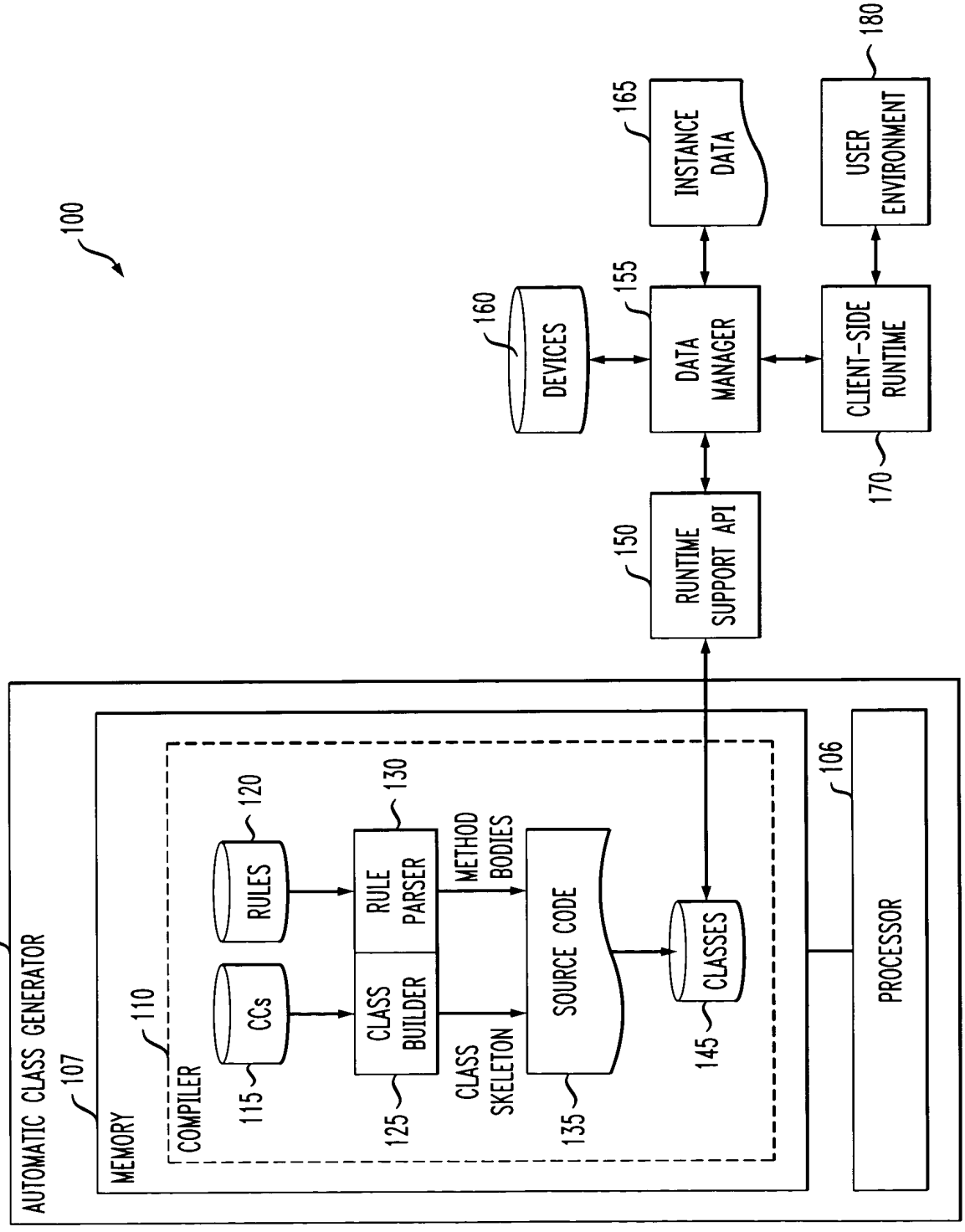

FIG. 1

FIG. 2

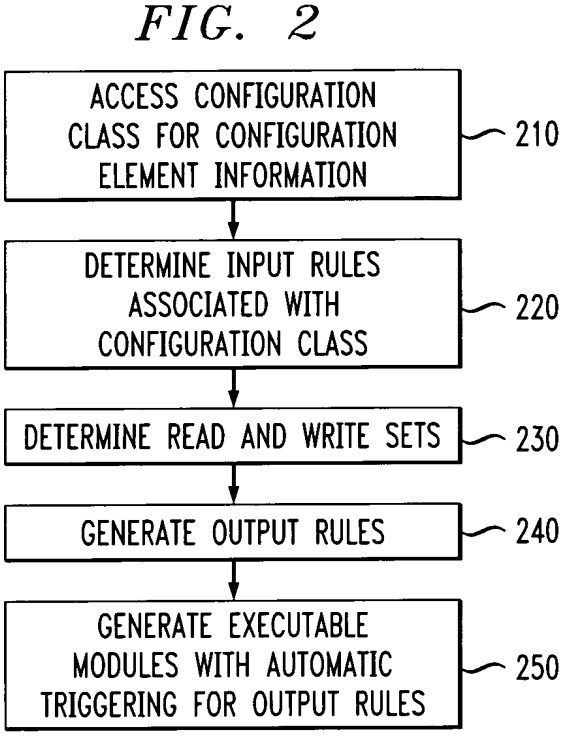

FIG. 3

```
<MOC className="SiteData" MOCVersion="1.0">
    <MOCElement name="server ID">
        <Type>
          <Int32 minValue="1" maxValue="100"/>
        </Type>
    </MOCElement>
    <MOCElement name="serverName">
        <Type>
          <String defaultValue=""/>
        </Type>
    </MOCElement>
    <MOCElement name="serverType">
        <Type>
          <EnumRef name="ServerTypeEnum"/>
        </Type>
    </MOCElement>
    <MOCElement name="serverAttr">
        <Type>
          <Attributes className="ServerComplexType" MOCVersion="1.0"/>
        </Type>
    </MOCElement>
</MOC>
```

FIG. 4

```
<Typedef name="PnniAtmAddr"> <!-- a predefined type -->
    <Type>
        <OctetString maxLength="20" minLength="0" />
    </Type>
</Typedef>
<!-- A type defined in terms of another type -->
<Typedef name="ArrayedAtmAddr">
    <Type arrayLength="10">
        <UserDefType name="PnniAtmAddr"/> <!-- type defined above -->
    </Type>
</Typedef>

<!-- this is inside a MOC that uses the above type definition -->
<MOCElement name="atmAddress" visibility="public">
    <Type arrayLength="25">
<UserDefType name="ArrayedAtmAddr"/>
    </Type>
</MOCElement>
```

FIG. 5

```
<MOC className="SiteData">
    <ExplicitRule name="iEMS_defaultValueRule"><![CDATA[
        public void iEMS_defaultValueRule()
        {
          serverID = 100;
          serverName = "HCFS1";
          serverType = D_serverType_hcf;
...
        }]]>
    </ExplicitRule>
      <MOCElement name="serverName">
          <ImplicitRule name="iEMS_validationRule"><![CDATA[
                public void iEMS_validationRule()
                {
                  // details of this rule are shown later.
                }]]>
          </ImplicitRule>
      </MOCElement>
    ...
</MOC>
```

FIG. 6

```
public class SiteData_1_0 {
    public int serverType;
    public int server ID;
    public ServerComplexType_1_0 serverAttr = new ServerComplexType_1_0(this, "serverAttr");
    public String serverName;
    ...
  }
```

FIG. 7

```
public class SiteData_1_0 //the 1_0 is the version
{
  ...
  public String serverName_get() {
    return this.serverName;
  }
  public int serverType_get() {
    return this.serverType;
  }
  public ServerComplexType_1_0 serverAttr_get() {
    return this.serverAttr;
  }
  public int serverID_get() {
    return this.serverID;
  }
  ...
}
```

FIG. 8

```
public class SiteData_1_0
{
  ...
  public void serverAttr_set(ServerComplexType_1_0 v) throws Exception {
    this.serverAttr = v;
    addToUpdateList("serverAttr", v, null);
  }

  public void serverName_set(String v) throws Exception {
    this.serverName = v;
    addToUpdateList("serverName", v, null);
  }
  ...
}
```

FIG. 9

```
public class SiteData_1_0
{
  ...
  public void serverName_update(String v) throws Exception {
    this.serverName = v;
    addToUpdateList("serverName", v, null);
  }

  public void serverAttr_update(ServerComplexType_1_0 v) throws Exception {
    this.serverAttr = v;
    addToUpdateList("serverAttr", v, null);
  }
  ...
}
```

FIG. 10

```
<MOCElement name="serverName">
  <ImplicitRule name="iEMS_validationRule"><![CDATA[
      public void iEMS_validationRule()
      {
      if(serverID < 1 || serverID > 250)
          throw new RuntimeException("serverID length error");
      if(serverType != D_serverType_hcf)
          throw new RuntimeException("serverType error");
      }
    ]]>
  </ImplicitRule>
</MOCElement>
```

FIG. 11

```
public void serverType_set(int v) throws Exception {
    //validate range -- omitted for space
    this.serverType = v;
    addToUpdateList("serverType", new Integer(v) , null);
    addTrigger(this, "serverName_iEMS_validationRule");
}
public void serverID_set(int v) throws Exception {                    } 1110
    //validate range -- omitted for space
    this.serverID = v;
    addToUpdateList("serverID", new Integer(v) , null);
    serverID_iEMS_modificationRule();
    addTrigger(this, "serverName_iEMS_validationRule");
  }

public void serverType_update(int v) throws Exception {
    //validate range -- omitted for space
    this.serverType = v;
    addToUpdateList("serverType", new Integer(v) , null);
    serverName_iEMS_validationRule();
}
public void serverID_update(int v) throws Exception {                 } 1120
    //validate range -- omitted for space
    this.serverID = v;
    addToUpdateList("serverID", new Integer(v) , null);
    serverName_iEMS_validationRule();
  }
```

FIG. 12

```
<MOCElement name="serverID">
  <ExplicitRule name="iEMS_modificationRule"><![CDATA[
        public void iEMS_modificationRule() throws IEMSException {
            try{          // dummy modification rules
                    if(serverID == 3)
                            serverName_set("Three");
                    if(serverID == 4)
                            serverName_set("Four");
              }
            catch(Throwable t){
                  throw new IEMSException(t);
            }
        }
]]>
  </ExplicitRule>
</MOCElement>

public NameValuePair[] triggerAlliEMS_modificationRule() throws Exception {
  NameValuePair[] values = new NameValuePair[1];
  serverID_iEMS_modificationRule();
  values[0] = new NameValuePair("serverID_iEMS_modificationRule", null);
  return values;
}
```

1210

1220

FIG. 13

```
<ExplicitRule name="iEMS_defaultValueRule"><![CDATA[
    public void iEMS_defaultValueRule()
    {
      serverID=100;
      serverName="HCFS1";
      serverType=D_serverType_hcf;
  ...
    }]]>

public void iEMS_defaultValueRule()
    {
      serverID=serverID_checkValue(100); //for range and integrity constraint
                                         //preservation
      //record this update to serverID -- omitted for clarity
      serverName=serverName_checkValue("HCFS1");
      //record update to serverName
      serverType=serverType_checkValue(D_serverType_hcf);
      //record update to serverType
    }
```

1310

1320